文章编号：1000-5889(2004)05-0101-03

LOTOS 规范风格在服务和协议设计中的应用研究

王继曾，张 键

(兰州理工大学 电气工程与信息工程学院，甘肃 兰州 730050)

摘要：分析了 LOTOS 规范语言的 4 种规范风格的特征和在分布式系统开发中的实际应用，描绘了网络服务和协议的基本设计属性，并讨论了规范风格在网络协议设计中的作用．最后将此方法应用于 AB 服务和协议的描述．

关键词：FDT；LOTOS；规范风格；协议

中图分类号：TP311 **文献标识码**：A

Investigation of application of LOTOS specification styles in service and protocol design

WANG Ji-zeng, ZHANG Jian

(College of Electrical and Information Engineering, Lanzhou Univ. of Tech., Lanzhou 730050, China)

Abstract: The features of four specification styles in LOTOS specification language as well as its practical application in the development of distributed system are analyzed in detail, the basic design attribution of network service and protocol is described, and the role of specification style in network protocol design is also discussed. Finally, this method is used to describe the AB service and protocol.

Key words: FDT; LOTOS; specification style; protocol

网络协议的开发需要消耗大量的资金和时间，如何高效地设计一个高可靠性和高质量的协议是一个必须研究的关键课题[1,2]．在网络协议的设计中，形式描述技术(FDT)的应用是至关重要和必不可少的[3]．在分布式系统的设计中大量使用形式描述语言的经验表明，为形式规范确定一个合适的结构是非常重要的，而且是经常被低估的问题．它的解决对确保在不同的抽象等级所必须开发的各种设计的质量具有重大的意义．因此研究描述规范风格在不同抽象等级中的应用可以很好地解决这个问题[4]．

1 LOTOS 概述

LOTOS 作为被 ISO 标准化的形式描述技术之一，主要用来设计分布式系统，特别是为 OSI 服务和协议．现在 LOTOS 已经被广泛用来描述大型数字通信系统．LOTOS 是通过在事件门径产生的可观察的事件之间定义临时关系来描述系统．LOTOS 有两部分组成：进程代数学和数据代数学．LOTOS 在数学上定义非常明确并且可表达性特别强，它可以描述并行协作的、非确定性的、同步的或异步的通信系统[5~7]．

2 LOTOS 的规范描述风格及其应用

形式描述技术 LOTOS 共有 4 种描述规范风格[4,5]，可应用于对协议和服务的描述，这些风格根据不同的目的，内含等级抽象、组件分解和重组的结构化思想，特别适于定义进程结构．

2.1 整体的风格(Monolithic style)

整体的风格在规范描述中无隐藏事件和并行操作符．描述规范表面上看起来是事件序列之间的分支选择．此规范风格禁止使用并行操作符，因此描述者不能对此描述规范实施功能性分解．如果把一个系统定义为一个简单的黑盒，使用整体的风格描述是非常有用的．它也能为其他描述风格描述的规范提供一个有用的出发点．尽管如此，通常在实际应用中很少用此风格描述更加复杂的分布式系统，因为它缺少结构，不适于人们理解和推理这些系统．因此，它更适于描述具有简单功能的实体，例如服务规范．

收稿日期：2003-10-22
基金项目：甘肃省自然科学基金(32204)
作者简介：王继曾(1950-)，男，山东招远人，副教授．

2.2 面向约束的风格(Constraint-oriented style)

在用此风格描述的规范中仅把可观察的交互表述出来,但是它们的时间序列被定义为不同约束的联合.此风格并不禁用并行操作符,可以产生比整体的风格更加紧凑和易于理解的描述规范.但是由并行操作符构造的实体不能被认为是软件实体,更确切地说是可被系统执行的可能事件序列的约束.面向约束的分布式系统的规范结构标准是从远端约束中分离出本地约束.使用这种方法作为实现轨道的前奏设计是非常有优越性的,因为分布式系统的不同部分通常分布在不同的实现单位.此风格允许在更低级的抽象层面中重用规范中的约束.另外,这些分离的约束具有互不相关的功能性.

2.3 面向状态的风格(State-oriented style)

用面向状态的风格描述的系统被认为是一单个的资源,它的内部状态空间被明确定义.因此,此种风格仅表示可观察的交互事件和被这些交互所处理的状态空间.这些交互事件同属于可选择的序列集,除此之外都是各自独立的.这种风格可以与面向资源的风格结合产生一种混合的风格,特别适用于通信协议后期阶段的设计.面向资源的风格通过一个资源集描述一个系统功能的分布状态,而在每个资源中使用面向状态的风格提供了此资源的功能信息.每个资源的面向状态的风格描述可以被用来开发这些资源的实现.然而,就分布式系统的抽象规范而言,由于此种风格自身的无结构性以及分解系统状态空间的复杂性,在使用上受到了很大限制.

2.4 面向资源的风格(Resource-oriented style)

在此风格中无论是可观察的还是内部的交互事件都可被描述出来.就可观察的交互而言,其行为可被定义为分离的内部交互隐藏的资源合成,这些资源可用任意风格来描述.

面向资源的风格把一个系统描述为通信资源的一个合成.可观察的、内部的和隐藏的事件都被包括进来,资源的同步通信通过门径来进行.这种风格特别适于面向于实现的规范描述,因为每个资源表现为一个自包含的实体.在每个资源中,任意描述风格可以被使用,包括面向资源的风格自身的迭代使用.用这种方法,可以通过分解资源来形成一个功能分解序列,直到分解后的资源足够简单到可以用任何风格来描述.

3 服务、协议描述和规范风格在设计中的作用

在网络协议设计中,假定存在形式定义的初始服务规范作为输入规范,并且可以把它转换为协议规范.

迭代的设计轨道,是基于重复的构件分解,可以被用来完成一个最终的协议规范.构件分解程序从 N 层服务规范开始,产生 N 层协议规范,这个协议规范根据 N 层协议实体和潜在的 $N-1$ 层服务构建.一个协议是一个服务的明确分解,也就是说,它的规范包括协议实体功能性的精确设计选择和潜在的服务.这个潜在的 $N-1$ 层服务可以以 N 层服务一样的方式被分解.这个过程一直被重复下去,直到这个潜在的服务可以通过实现组件被直接实现为止.这些实现组件属于最下面的底层,可以匹配所需的行为.图 1 描述了分层的协议设计.

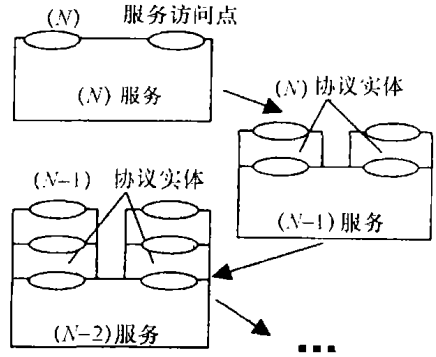

图 1 分层的协议设计

Fig.1 Layered protocol design

从不同的观点来看,一个服务和一个相应的协议是同一分布式系统行为的可供选择的规范.服务是系统的一个整体描述,根据可观察的事件(用 OSI 术语来讲,是服务原语),最好用一组分离约束来描述.这些约束可以根据本地约束和远端约束来构造.本地约束也就是服务访问点中相关的本地行为部分;远端约束就是不同服务访问点之间的相关行为部分.这种描述形式就是面向约束的风格,它是最常用的描述服务的风格.

一个协议是一个系统的分布式描述,也就是说,它根据对象或资源来描述系统.面向资源的风格是最合适的描述协议规范形式.在面向资源的规范中,设计的结构被描绘为进程,一个进程代表一个对象或资源.资源之间的内部通信被隐藏在系统环境当中.

协议实体实现了服务的分布.因此服务的本地约束可以在协议实体的定义中再次出现,规范中早期定义的元素可以重用,简化了以后的验证.服务的远端约束可以被分解成协议实体和潜在的服务.图 2 描绘了规范风格在表达服务和协议规范中的应

用.

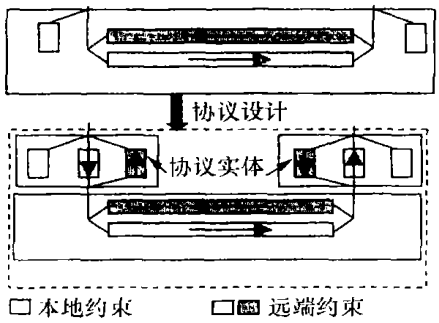

图 2　用规范风格将服务分解成协议

Fig.2　Decomposition of service into protocol using specification styles

一个规范风格是一个特定的根据在设计进程中的服务目的构造规范的方法.在一个设计进程的初始阶段,需要一个强调需求捕获和避免实现策略的风格,这个风格可以是面向约束的风格或整体的风格.在一个设计进程的最终阶段,需要一个强调实现的真实结构的风格,这个风格可以是面向资源的风格或者面向状态的风格.由于面向整体的和面向状态的风格模糊了规范结构,因此很少使用,除非在一些比较简单的行为描述中.在实践中,这些风格在定义一个进程中经常混合使用.哪个风格被应用依赖于抽象等级和设计环境.

4　AB(alternating bit)服务和协议的描述

AB 协议是最简单的协议[2,3,8].以此为例,说明 LOTOS 规范风格在其描述中的应用.图 3 描绘了 AB 服务和协议之间的关系.此例仅考虑单向传输.

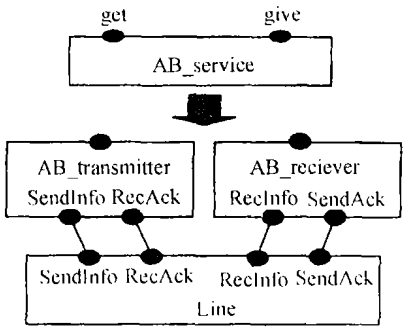

图 3　AB 服务到协议的转换

Fig.3　Transformation from AB Service to AB protocol

由于服务极其简单,可用整体的风格描述其服务规范,如下所示:

```
Specification AB-service[get, give]:noexit:=
Behaviour
    AB-service[get,give]
Where
    Process AB-service[get,give]:noexit:=get ? data:bitstring; give ! data; AB-service[get,give]
    Endproc
Endspec
```

AB 协议是 AB 服务的实现.协议已经非常简单,不须先用面向约束的风格构造规范,再用面向资源的风格进行构件分解.因此可直接用面向资源的规范风格构造模型,该模型包括 3 个协议实体:发送实体、通道实体和接收实体.

```
Specification AB-protocol [get,give]:noexit
Behaviour
Hide tout,send,receive in
    ((AB-transmitter[get,send,receive](0)|||AB-receiver[give,send,receive](0))|[send,receive]|line[send,receive])
Where
    (* process definitions AB-transmitter, AB-receiver and line *)
Endspec
```

以上规范是在 AB 协议的最高抽象层上的设计描述,不妨设为第 N 层抽象,那么第 $N-1$ 抽象层的规范是对各实体的描述,通常以进程(AB-transmitter、AB-receiver 和 line)的形式出现.由于这些实体的结构已经相当简单,因此都可以服务规范的形式出现,即使用整体的风格描述.

参考文献:

[1] 罗铁庚,陈火旺,齐治昌,等.协议形式化开发环境的规范语言[J].软件学报,1997,8(11):817-822.

[2] 龚正虎.计算机网络协议工程[M].长沙:国防科技大学出版社,1993.

[3] 古天龙,蔡国庆.网络协议的形式化分析与设计[M].北京:电子工业出版社,2003.

[4] Vissers C A,Scollo G,Sinderen M V, *et al*.Specification styles in distributed systems design and verification [J].Theoretical Computer Science,1991,89:179-206.

[5] Gomez S P.The lotosphere design methodology:Guidelines [R].Twente:ESPRIT 2304 Lotosphere Project,1992.

[6] Bolognesi B,Brinksma E.Introduction to the ISO specification language LOTOS [J].Computer Networks and ISDN Systems,1987,14:25-59.

[7] ISO/IEC 8807 — 1989,Information processing systems,open systems interconnection,LOTOS —— A formal description technique based on the temporal ordering of observational behaviour [S].

[8] Hart N.Protocol validation and implementation:A design methodology using LOTOS and ROOM [D].Ottawa:University of Ottawa,1998.